

修士論文概要書

2010 年 2 月提出

学籍番号 5108B009-1 CD

専門分野	情報理工学専攻	氏 名	板橋 裕介	指 導 教 員	大附 辰夫	印
研究指導	情報アーキテクチャ研究					
研 究 題 目	アドホックネットワークにおけるクラスタの接続性と クラスタヘッドの負荷分散を考慮したルーティングに関する研究					

1 はじめに

近年、モバイル通信端末の普及に伴い、既存の通信インフラを介さずに多数のノードが相互に通信する、モバイルアドホックネットワークが注目されている。アドホックネットワークでは、直接には電波の届かない遠隔のノード同士をマルチホップ型の通信形態で接続するが、各ノードは自由に移動してネットワークトポロジは絶えず変化するので、通信経路を制御するためのルーティングプロトコルが重要な課題である。一般的なルーティングプロトコルでは、大規模アドホックネットワークにおいて経路構築のための制御パケットの溢れ問題が指摘されている。この問題点を解決するために、クラスタリングを用いたルーティング手法が研究されている。ネットワークを複数のクラスタに分割することで、フラディングによる無駄な制御パケットを大幅に削減できる。クラスタヘッドは接続関係の情報を管理する必要があるため、ネットワーク全体に流れるトポロジ情報を削減するためにクラスタの総数を抑制することが望まれる。また、ノードの移動によるトポロジの変更に伴いクラスタの再構成が行われるが、クラスタヘッドが変更されると、新しいクラスタヘッドは新たにクラスタメンバの情報を収集しなければならないため制御パケットが増加する原因となる。よって、ネットワークにおけるクラスタの総数を小さくし、クラスタヘッドの変更数を小さくすることで、安定した接続性の高いクラスタを構成できる。クラスタの接続性を向上させるためにはクラスタヘッドの選択方法が重要である。既存手法に隣接端末数が最大のノードをクラスタヘッドと選択するものがある。ネットワーク全体におけるクラスタ数は削減できるが、ノードの移動に伴い隣接端末数は頻繁に変更されるので、クラスタヘッドの変更数は増加する。また、クラスタリング手法では必ずクラスタヘッドを経由した経路を構築するために、ホップ数の増大やクラスタヘッドの負荷によるパケット到達率の低下などの問題がある。

本論文では、クラスタの接続性とクラスタヘッドの負荷を考慮したルーティングプロトコルを提案する。提案手法（CLBC）では、クラスタの接続性を高めるためにクラスタヘッドの選択方法を2ホップ先のノードも含めて隣接端末の最も多いノードとすることで、ネットワーク全体におけるクラスタの総数を削減し、かつクラスタの再構成によるクラスタヘッドの変更回数も削減している。またクラスタヘッドを経由する必要がないときは最短の経路をとることで、ホップ数の削減とクラスタヘッドの負荷を分散させている。

2 提案手法

提案手法の概要を述べる。

クラスタの接続性 クラスタの構成手順を図1に示す。

Step1 クラスタヘッドはClusterControl (CC) パケットをブロードキャストする。CC パケットの送信範囲を k ホップと限定することで、1つのクラスタの大きさが決まる。
Step2 CC パケットを受けたノードが、クラスタに所属していないノード、または他のクラスタのクラスタヘッドで CC パケットの送信元よりも隣接端末が少ないノードならば、CC パケットの送信元ノードをクラスタヘッドとするクラスタのメンバになる。
Step3 クラスタヘッドにControlReply (CR) パケットを返送。
Step4 CR パケットを受けたクラスタヘッドは、メンバリストを更新する。
Step5 クラスタヘッドよりもメンバノードの方が隣接端末数が多い場合には、クラスタヘッドはそのメンバノードにControlReplyAck (CRA) パケットを返送する。同時に、クラスタヘッドから一般ノードに変わったため、全てのメンバノードに対してエラーメッセージを送信する。
Step6 CRA パケットを受信したノードは、新しいクラスタヘッドとなる。

図 1: クラスタ構成手順。

本手法では、クラスタの総数を抑えつつクラスタヘッドの変更数を削減するために、単純な隣接端末数でなく2ホップで隣接する端末数も考慮している。1ホップだけの隣接端末数では局所的なノードの接続しかわからないため、ノードの移動により隣接端末数が大きく変動しクラスタヘッドの変更につながりやすいが、2ホップで隣接する端末数まで考慮することで大域的に接続性を判断できるので、ノードの移動に対してもクラスタの接続性を保ちやすくなりクラスタヘッドを変更せずに済むようになる。クラスタを構築するために用いる制御パケットに隣接するノードのID情報を付加しており、制御パケット数を増やすことなく、2ホップ先のノードを確認している。図2において、ノード5がクラスタヘッド2へ制御パケットを送信する場合を例とする。ノード5は隣接しているノードのID情報(2, 3, 4, 6)を制御パケットに付加して送信する。制御パケットを受信したクラスタヘッド2は、そのID情報と自身の隣接ノードのID(1, 4, 5)を比較する。ノード3, 6のように制御パケット受信側のノードには隣接していないが、制御パケット送信側のノードにのみ隣接しているノードを2ホップで接続していると、クラスタヘッド2は判断している。このとき、ノード4は両方の隣接ノード情報に含まれているため、クラスタヘッド2はノード4と制御パケットを送信したノード5は直接通信が可能と判断でき、次で述べる経路構築にも応用できる。

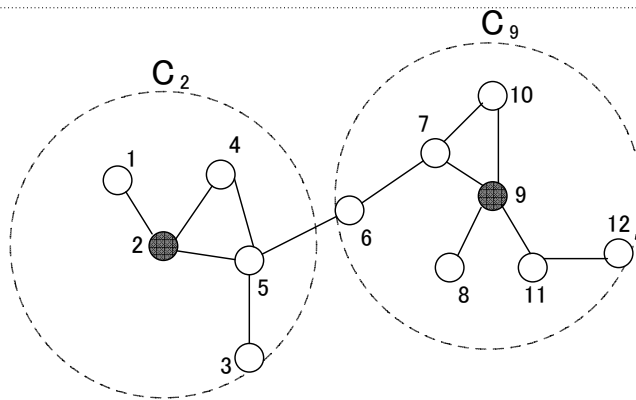


図 2: クラスタリング.

クラスタヘッドの負荷分散 本手法では、クラスタヘッドを通る必要がない場合にはクラスタヘッドを通らない経路を構築してホップ数を削減し、クラスタヘッドにかかる負荷を分散させている。ここで、クラスタリングを利用した経路探索の手順を図 3 に示す。

Step1 メンバノードはクラスタヘッドに RREQ パケットをユニキャストで送信。
 Step2 RREQ パケットを受信したクラスタヘッドは、自身のクラスタのメンバノードに宛先ノードが存在するかを確認し、存在するならば宛先ノードへ RREQ パケットをユニキャストで送信。
 Step3 自身のクラスタ内に宛先ノードが存在しなければ、RREQ パケットを隣接するクラスタへ転送。
 Step4 Step1, 2, 3 を宛先ノードが見つかるまで繰り返す。
 Step5 RREQ パケットを受信した宛先ノードは、RREP パケットを経路の逆順に返送。

図 3: 経路構築手順。

構築した経路においてクラスタヘッド C に隣接する 2 つのノード A, B があり、A, B も隣接しているとき、クラスタヘッドを経由せずに直接通信することでホップ数とクラスタヘッドにかかる負荷を削減できる。メンバノード同士が直接通信可能かどうかは、クラスタヘッドから 2 ホップ先のノード情報を集める時に判断できる。例えば図 5 において、ノード 2 からノード 8 に通信する場合を考える。クラスタリングの経路探索のため必ずクラスタヘッドを経由して 2, 1, 4, 5, 6, 8 という経路を構築するが、クラスタ 1 はノード 2, 4 が直接通信可能と分かっており、また同様にクラスタ 6 においてもノード 5, 8 が隣接していることを知っているの、クラスタヘッドを通らない 2, 4, 5, 8 という最短経路を構築できる。

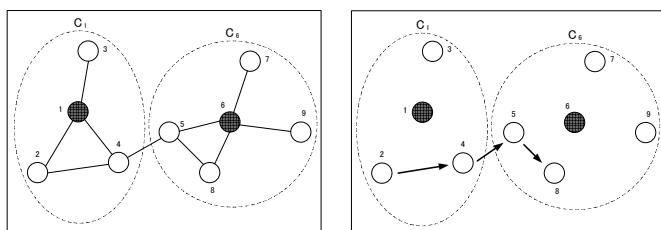


図 4: クラスタヘッドを経由しない経路構築。

3 シミュレーション実験

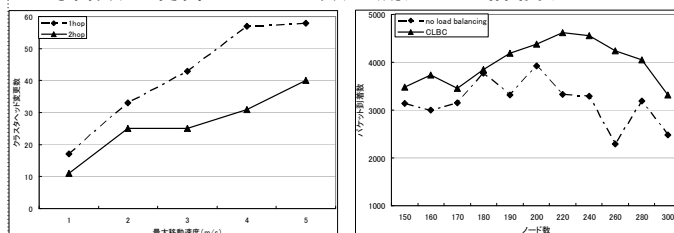
提案手法の有効性を示すために、コンピュータによるシミュレーション実験の評価を行う。評価に当たってはシミュレータとして ns-2 を用いた。

実験 1: クラスタの接続性 ノードの移動速度を速くすると、トポロジの変化が大きくなり、クラスタの再構成によるクラスタヘッドの変更回数が増加する。よって、ノードの最大移動速度を変化させたときに、クラスタヘッドの変更回数を測定して評価する。

実験 2: クラスタヘッドの負荷分散 ノードの移動速度を歩行者程度の速度とし、ノード数を変化させたときに、パケット到着数と通信経路のホップ数を測定して評価する。

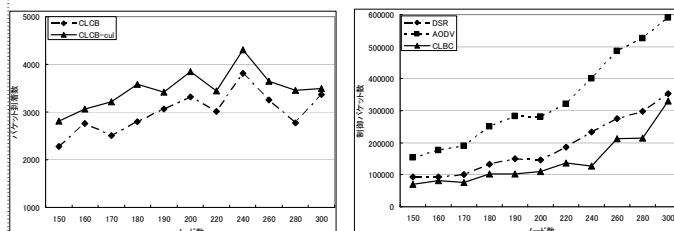
実験 3: 片方向リンク 提案手法を拡張し、片方向リンクを含む環境において、片方向リンクの検出して片方向リンクを避けることで双方向リンクのみの安定した経路を構築する。片方向リンクを生じさせるために 2 つの通信可能範囲のノードを混在させ、ノード数を変化させたときに、パケット到着数を測定して評価する。

実験 4: 既存手法 提案手法と既存のフラットルーティングを比較する。ノード数を変化させたときに、パケット到着数と制御パケット数を測定して評価する。



(a) 実験 1

(b) 実験 2



(c) 実験 3

(d) 実験 4

図 5: 実験結果。

4 おわりに

本稿ではクラスタの接続性とクラスタヘッドの負荷分散を考慮したルーティングプロトコル CLBC を提案した。実験結果により、2 ホップ先の隣接ノード情報を利用することによりクラスタヘッドの変更数は 45% 削減、クラスタヘッドの負荷分散により 26% のパケット到着数の向上と 17% のホップ数を削減、片方向リンクを考慮することによりパケット到着数を 28% 向上、既存手法と比較してもパケット到着数の向上と制御パケット数の削減を確認した。